# COOPERATIVE ANALOG AND DIGITAL (CANDI) TIME SYNCHRONIZATION PROTOCOL FOR LARGE MULTI-HOP NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Sunghwan Cho

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Compter Enginerring

Georgia Institute of Technology
December 2011

# COOPERATIVE ANALOG AND DIGITAL (CANDI) TIME SYNCHRONIZATION PROTOCOL FOR LARGE MULTI-HOP NETWORKS

Approved by:

Professor Mary Ann Ingram, Advisor
School of Electrical and Compter
Enginerring
*Georgia Institute of Technology*

Professor Edward J Coyle
School of Electrical and Computer
Enginerring
*Georgia Institute of Technology*

Professor Gregory David Durgin
School of Electrical and Computer
Enginerring
*Georgia Institute of Technology*

Date Approved: 11 November 2011

*To my wife and son.*

# ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who gave us the possibility to complete this thesis. First and foremost I would like to give my sincerest thanks to my thesis advisor Dr. Mary Ann Ingram for trusting me and offering the chance to work on this thesis. Her help, stimulating suggestions, constructive comments and encouragements have greatly improved this work. I consider myself lucky to be one of her students.

I want to thank my thesis committee, Dr. Coyle and Dr. Durgin for taking the time to review and critique for my work. I also would like to thank Republic of Korea Army, which gave me the financial support for two years and the great opportunity to study at Georgia Tech.

And last, but most importantly, I am deeply grateful to my families, whose love and guidance is always with me in whatever I pursue. It is to them that I dedicate this work.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

The objective of this thesis is to develop an improved time synchronization (TS) protocol for large multi-hop wireless sensor networks (WSNs). For such networks, it is well known that with conventional methods, such as Timing-sync Protocol for Sensor Networks (TPSN), Reference Broadcast Synchronization (RBS), and Flooding Time Synchronization Protocol (FTSP), TS error increases as the hop number increases. To reduce the number of hops to cover the large network and exploit the spatial averaging of TS error between clusters, a novel method combining Concurrent Cooperative Transmission (CCT) and Semi-Cooperative Spectrum Fusion (SCSF) is proposed.

This novel method, named Cooperative Analog and Digital (CANDI) Time Synchronization Protocol, consists of two phases: The digital stage and the analog stage. The digital stage uses CCT to broadcast a TS packet, wherein cooperating nodes simultaneously transmit the same digital TS packet in orthogonal channels that experience independent multi-path fading. Each receiver is capable of combining the differently faded copies, thereby achieving a significant SNR advantage, through array and diversity gains. In the analog stage, the cooperating nodes simultaneously transmit their individual estimates of the time, encoded across orthogonal dimensions. Nodes receiving this signal combat fading and reduce estimation error in one step through the averaging inherent in diversity combining.

Simulations using MATLAB for line and two-dimensional networks with respect to various parameters are used to evaluate the performance of CANDI. The simulation result proves that the TS error of CANDI is significantly lower than TPSN, which is one of the typical TS methods in WSNs.

# CHAPTER I

# INTRODUCTION

With the recent advances in micro electro-mechanical systems (MEMS), wireless communications, and digital electronics, low-cost, low-power and small-sized sensing devices can be possible. Using these sensing devices, many researchers are trying to make "smart environments," where sensors, actuators, displays, and computer elements are interwoven into the everyday experience of our lives, and connected to a continuous network [9]. The sensing nodes, which are the basic component of "smart environments," measure various physical features, such as temperature, pressure, humidity, or location of object. Then they transmit the raw data or the data calculated simply at the sensing device to the sink node, forming so-called wireless sensor networks (WSNs). Due to the intrinsic characteristics of sensor nodes, such as collaboration among nodes, distributed topology and close-adjacency to the objective physical feature, WSNs have the advantage of capturing the physical features more reliably and efficiently than the traditional sensing architectures [5]. Also, the fact that the sensor nodes may have various types of sensors enables the WSNs to have a wide range of applications. The potential applications of WSNs include:

- Monitoring Applications: the monitoring of wild sensitive habitats without interferences using WSNs helps the biologist to observe them as they are [29]. WSNs are also used to observe natural phenomena that people cannot approach, such as hurricanes, volcanoes and forest fires [26]. Many public infrastructures, such as bridges and buildings, can also be remotely inspected through WSNs [17].

- Military Applications: sniper detection system [1], battlefield monitoring system [22], and target tracking system [21] are examples of military applications.

- Smart Home and Office: home appliances with WSNs, such as vacuum cleaners, microwave ovens, refrigerators, and DVD players can interacte with each other and connect to the external network [4]. Controlling the light and temperature with wireless sensors and actuators can save energy consumption in the office. [28].

- Health Applications: this category includes telemonitoring the status of patients, tracking and monitoring of doctors, drug administrators, and patients inside a hospital, [34].

WSNs are usually distributed over a large area, and the sensor radios have only a short range. Multi-hop communication can reduce the cost of such a network by not requiring that every sensor be within one hop of the higher-functioning and more expensive sink nodes. Multi-hop communication presents additional challenges to the typical wireless communication protocol. For example, synchronizing the clocks on all nodes in the network, when there is no external reference, is more difficult than for a star topology network [4].

## *1.1 Problem Statement*

The time synchronization (TS) is the procedure to distribute the common notion of time across some or all the nodes, and then make the local clocks have the same time reference [5]. In WSNs, to coordinate the multiple data from the sensor devices, the TS is a crucial part of collaborating between the sensing nodes. As a result of TS, the WSNs can have following functions [5, 33]:

- Event Ordering: after the sensor devices sense an event, they send the sensed data to the sink node for the fusion. However, the delays between the sink and

the sensor devices vary according to the network environment (hop number, propagation time, and network load), the sink node cannot order the event only according to the arrival order of the data. However, the timestamp with the sensed information can be used to reorder the event easily.

- Data Fusion: Data fusion is a fundamental operation in WSNs to process the collected data into meaningful information. It requires some or all the nodes in the WSN to have a common time scale.

- Power Management: the energy efficiency is a crucial part to extend the lifetime of WSNs. Using the synchronized time across the network, the node can make its duty cycle (scheduling sleep and wake-up mode) consistent with other nodes, so that they save huge amounts of energy.

- Synchronized Network Protocol: the clock synchronization enables time division multiple access (TDMA), which is a transmission scheduling protocol. In TDMA, the nodes need to follow the common time frame, thus the synchronized time between the nodes is required.

A well designed TS protocol must address other requirements, such as low-cost clock, low-energy consumption, and short-time occupancy of wireless communication channel. Also, the wide variety of WSN applications imposes difficulty on making a standard TS method. Therefore, in WSNs, it should be a goal of TS development to find the optimal point between the accuracy of TS and the given budgets.

Many TS protocols for WSNs, which will be reviewed in Chapter II, are well studied and implemented on the WSNs, however they have a common problem that the TS error accumulates as the hop number increases [15]. This problem is worse in the large multi-hop WSNs, such as in the early warning flood detection and the battlefield monitoring system. Even worse, since the large area of WSNs usually require a large number of nodes, the large number of nodes results in the long-time

occupancy of the communication channels only for performing TS, if the TS method is based on the single-node-to-single-node broadcast. Therefore TS would degrade the comprehensive throughput and the energy efficiency of WSNs.

## *1.2 Contribution*

In this thesis, we present a novel TS protocol that is suitable for large multi-hop networks. It combines two existing types of cooperative transmission (CT), a digital (or decode-and-forward) one called Concurrent Cooperative Transmission (CCT) and an analog one called Semi-Cooperative Spectrum Fusion (SCSF). CT is a physical layer wireless communication scheme that exploits the SNR advantage in a receiver by combining transmissions from multiple transmitters. The new TS method is named Cooperative Analog and Digital (CANDI) TS protocol. By using the advantage of range extension of CT, CANDI is able to reduce the number of total hops to cover the large multi-hop networks, so that CANDI improves the overall TS error, compared to other typical TS methods based on non-CT. To the best of our knowledge, this CANDI TS algorithm is the first TS protocol that the CT is applied for the range extension. (The firefly or pulse-coupled oscillator (PCO) concept [13, 32] are similar to our proposed method in that the nodes behave simply and autonomously and transmit in large groups, but the process of synchronization does not depend on CT range extension.)

## *1.3 Outline of the Thesis*

The remaining part of the thesis is structured as followed. In Chapter 2, we briefly review the properties of the clocks in WSNs, and describe some of typical TS protocols. Then, the recent developments of CCT and SCSF will be introduced. Chapter 3 describes the CANDI protocol in detail and explains how CCT and SCSF have benefits while performing the TS procedure in large multi-hop network. The simulation result for line networks is given comparing with TPSN. In Chapter 4, the CANDI

in two-dimensional (2-D) networks is analyzed and its performance is shown through simulation.

# CHAPTER II

# RELATED WORK

In this chapter, we introduce the characteristics of the local clock in a sensor device and the sources of time synchronization error. We also introduce some of the well-known TS protocols for WSNs. Then we review the recent developments of CCT and SCSF.

## *2.1 Clock Synchronization*

Clock synchronization has been studied by many researchers for many years, and many TS protocols have been proposed so far. As introduced in Chapter I, sensor network applications need clocks for determining the duty cycle, scheduling tasks, and combining collected data to meaningful information.

In centralized systems, each node can get the time by requesting the issue of time from the kernel, such as Network Time Protocol (NTP) [23] server. However, in distributed systems like WSNs, there is no global time server. While GPS can provide high quality TS in some applications, the GPS signal is often not available at sensor locations, such as indoors, or costs too much energy. Thus, each node has to be equipped with its local clock, and the distribution of time among the sensor nodes, i.e., a TS procedure, is necessary, However, the characteristics of clocks equipped at the sensor device and the deterministic and random delays make the TS process challenging.

### 2.1.1 Time and Clock

There are various reasons why the clocks of sensor nodes have different notions of time. The sensor devices might be turned on at different times, so that the clocks

have arbitrary clock offsets. Also, a quartz clock is usually used in sensor devices to satisfy the price requirement of sensor node; these clock's behaviors, especially clock rate, are subject to change due to the environmental conditions or other external effects such as temperature, atmospheric pressure, voltage changes, and hardware aging [33]. Hence, the relative time difference among the local clocks keeps changing as time elapses, even though the local clocks might be synchronized within the required accuracy immediately after the TS is performed. For these reasons, the TS has to be performed periodically or when the TS procedure is requested by the node. These are called continuous synchronization and on-demand synchronization, respectively.

### 2.1.2 Time Synchronization Error

Assume that Node B wants to be synchronized to Node A. Node A transmits its time information to Node B. If there is no delay while the time information is being delivered, Node B can synchronize to Node A just by setting its local clock to the received time information. However, in reality, there are various delays during the message delivery, which makes the TS process problematic. The various components of TS error include the following delays [11, 33]:

- Send time: When the node decides to transmit a packet with its time, the packet is made at the application layer, then passed to the MAC layer for transmission. This delay also includes the software delays introduced by underlying operating system.

- Access time: After the packet arrives to the MAC layer, it has to wait until the channel is available. This delay is usually considered as the most significant component and quite variable depending on the specific MAC protocol.

- Transmission time: This is the actual time for transmitting the packet in the Physical layer. The absolute value of this error is deterministic and negligible as compared to other sources [11].

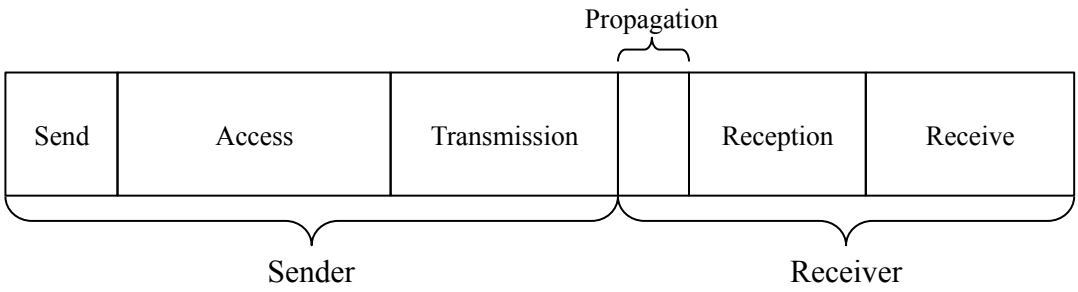



Figure 1: Decomposition of TS error

- Propagation time: This refers to the actual time taken by the packet to be transmitted in the wireless link from the sender to receiver. This time is a nearly deterministic function of the physical distance between the sender and the receiver [22].

- Reception time: This is the time taken by the receiver to receive the bits and pass them to the higher layer, which is the opposite of the transmission time.

- Receive time: When the bits have arrived at the receiver, these bits are composed into a packet, which is passed to the application layer where the packet is processed. Receive time is the delay taken in this whole activity.

In Figure 1, the different sizes of boxes gives an intuition about the absolute value of each component [11], even though it does not provide the actual ratio. The delay components can be categorized into fixed portion (deterministic) and variable portion (random). The variable portion depends on various network parameters, such as the type of sensor nodes, network environment, and network traffic. To model the random components, probability density functions (pdf), such as Gaussian, Exponential, Gamma, Weibull, and Log-normal [25, 20, 6] pdfs, are used.

## *2.2 Traditional Time Synchronization Protocols for WSNs*

Many applications of wireless sensor networks (WSNs) require accurate time synchronization (TS) at each node so that data measured in different areas of the network can be properly time-tagged and later processed (e.g., correlated) at a central location. Also, with these synchronized times, the nodes can be more precisely scheduled to transmit their data, so that the energy and channel efficiency can be improved. Moreover, synchronized time among sensor nodes also can be used for the purpose of security, localization, and tracking protocols [33].

Many TS methods for WSNs have been studied so far. TS methods for WSNs can be divided into 2 categories: centralized TS and decentralized TS. Several popular centralized TS methods include Reference Broadcast Synchronization (RBS) [10], Timing-sync Protocol for Sensor Networks (TPSN) [11], and the Flooding TS protocol (FTSP) [22]. The typical decentralized TS method is the Pulse Coupled Oscillators (PCO) protocol [13]. These are described in detail in this section.

### 2.2.1 Centralized Time Synchronization Protocol

The centralized TS protocol is that all nodes in the network set their local clock to the reference node's clock by transferring the time information. There are three well-known types of transferring approaches, such as the receiver-receiver synchronization, sender-receiver synchronization (the two-way message exchange), and one-way message dissemination.

#### *2.2.1.1 Reference Broadcast Synchronization*

RBS is representative of the receiver-receiver schemes. The third party broadcasts a beacon that does not contain any timing information to all the receivers, then the receivers use the arrival time of the beacon only as a point of reference for comparing their clocks. By comparing their clocks to one another, the receivers calculate their relative phase offsets. This enables them to exclude the time uncertainty on

sender [10], which is illustrated in Figure 2. By reducing the critical path of error, the time synchronization error can be significantly decreased. However, since the synchronized nodes do not receive the absolute time, the additional time information exchange between the synchronized nodes should be needed. Moreover, the center node needs an additional time synchronization process for itself. Both facts result in increased energy consumption [31].

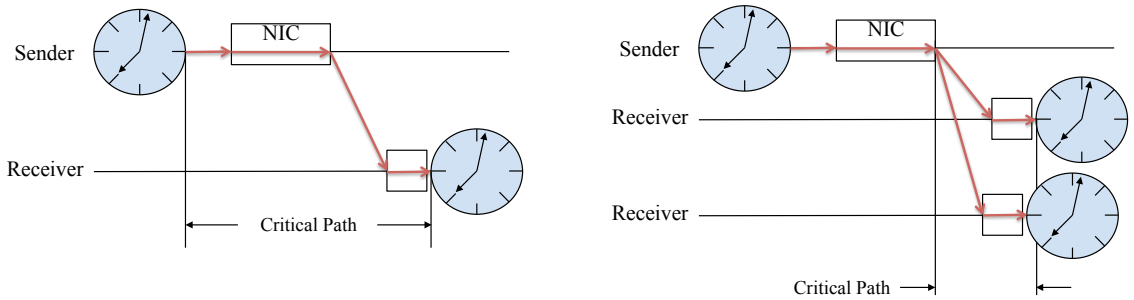



Figure 2: A critical path analysis for traditional time synchronization protocols (left) and RBS (right) [10]

#### *2.2.1.2 Timing-sync Protocol for Sensor Networks*

In contrast to RBS, TPSN [11] is representative of sender-receiver schemes. TPSN consists of two phases: *Level Discovery Phase* and *Synchronization Phase*. At the *Level Discovery Phase*, the wireless nodes are classified into several levels and at *Synchronization Phase*, the *Level i+1* node synchronizes its clock to the *Level i* by exchanging time stamps. Figure 3 illustrates the two-way messaging between a pair of nodes at the *Synchronization Phase*. The times T1, T2, T3, and T4 are all measured times. Node A transmits the TS packet at time T1 to Node B. This TS packet contains Node A's level and the time T1. Then, Node B transmits the acknowledgment packet containing T1, T2 and T3 at time T3 to node A. By using this time information, Node A can adjust its local clock to Node B's clock.

This scheme has the advantages that it can eliminate the propagation time error

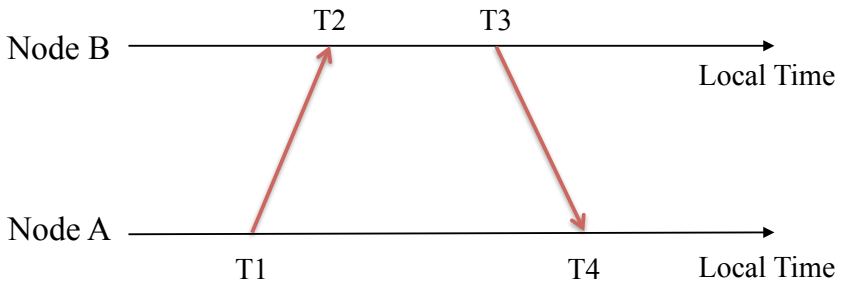



Figure 3: Two-way communication between nodes in TPSN [11]

by handshaking of timestamps, and the send time error can also be reduced by time-stamping the packet at the MAC layer rather than at the application layer. TPSN has 2x better performance than RBS [11] in the average timing accuracy. However, since TPSN is still 1:1 time synchronization scheme, it requires high battery consumption and fairly long times to synchronize the large networks.

#### *2.2.1.3 Flooding Time Synchronization Protocol*

FTSP synchronizes nodes by broadcasting the time information with a single radio message. This scheme saves the initial phase of establishing the tree and achieves high accuracy by time-stamping at the MAC layer and handling comprehensive error with linear regression. Even though it has advantages that it relatively consumes less energy and takes shorter time than previous two methods, FTSP has an error that grows exponentially with the size of the network [22], since FTSP is an open-loop TS method.

### 2.2.2 Non-Centralized Time Synchronization Protocol

#### *2.2.2.1 Firefly Synchronization*

Firefly Synchronization [32] is one of the non-centralized time synchronization methods. Every node has a phase function $\phi(t)$. This function evolves linearly over time $t$ as shown in Figure 4 (a). When the phase reaches a threshold $\phi_{th}$, the node emits a pulse of light. Therefore, if the node is isolated, the node emits a pulse periodically.

However, if the node receives a pulse from another node in the network, it increases its phase by up to $\Delta\phi(\phi)$ as shown in Figure 4 (b), which results in that the node emits a pulse earlier. This process makes eventually all nodes emit the pulse simultaneously, which means that nodes go to synchrony.

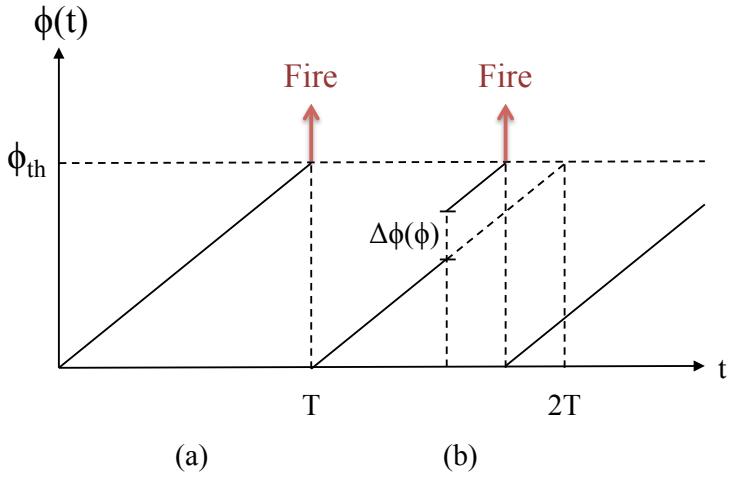



Figure 4: Time evolution of the phase function

This scheme does not have a reference node, therefore the scheme possesses robustness on the destruction of the reference node, and the time to synchronize the networks is in inverse proportion to the number of nodes [13]. However, if there are only a small number of nodes in the network, the protocol takes a long time to synchronize the network, so that it wastes much energy. Also, as in the experimental result of this method, there exists possibility that one network can be synchronized into different groups depending on the network topologies [24].

This is a cooperative transmission method, because it involves groups of nodes that transmit together, but the earliest node one to fire will be the one that impacts the receivers. This is a form of selection. However, the cooperation does not yield range extension, and therefore its effectiveness will diminish for large multihop networks.

## *2.3 Cooperative Digital and Analog Transmission*

In this section, two cooperative transmission schemes are described. One uses digital transmission and the other uses analog transmission. These two schemes are both utilized in the proposed TS scheme, which is the main contribution of the proposed research.

### 2.3.1 Concurrent Cooperative Transmission

Cooperative Transmission (CT) is a physical layer wireless communication scheme that exploits the SNR advantage in a receiver by combining transmissions from multiple transmitters. Concurrent Cooperative Transmission (CCT) is one of three types of CT methods, the other two being coherent beam-forming and time division CT (TDCT).

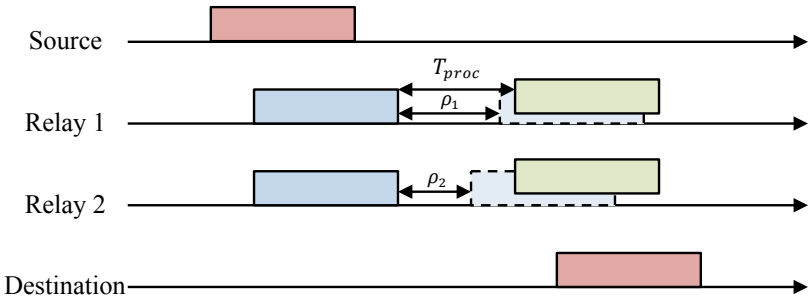



Figure 5: Packet timing

In CCT, the nodes in a cooperating cluster transmit the same digital message in independent orthogonal channels at the same time, so that the receiver treats the collection of signals as though they came from one array diversity transmitter. The transmitted signals are synchronized in time, based on a packet that all cooperating nodes received just previously. CCT was demonstrated experimentally in [8], in which the transmission time for each node is scheduled by allocating a fixed time $T_{proc}$ after the end of the packet just received. Figure 5 describes the role of $T_{proc}$ that enables the receivers to avoid the uncertainty of latency caused by random processing times

in Universal Software Radio Peripheral (USRP1) and GNU-Radio.

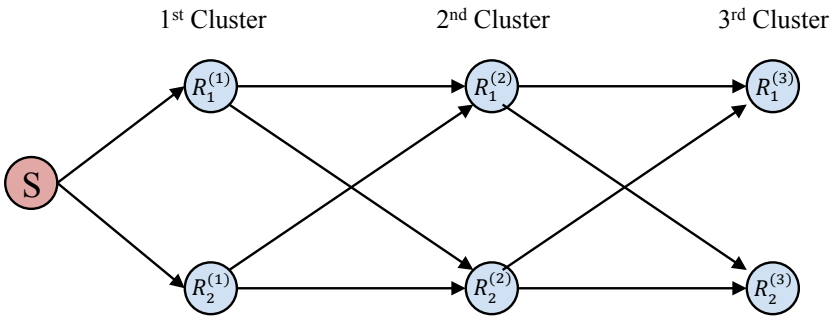



Figure 6: Simple CCT network

CCT can be used in succession, for the purpose of broadcasting [27, 18]. In this case, a source transmits, then all nodes that can decode correctly relay at their individually scheduled times. Then all nodes that *are able to* decode the CCT packet, become the next transmitting cluster, and so on. The simplest example of cascade CCT is illustrated in Figure 6. This broadcasting scheme, which is termed Opportunistic Large Array (OLA) broadcasting, was first treated theoretically and in simulation by Scaglione's research group [27], and later extended and experimentally demonstrated by Ingrams's research group [8, 7, 16]. The word "opportunistic" refers to the fact that the number of nodes participating in each CCT cannot be predicted beforehand. 90% of the rms transmit time spreads (RTTSs) are less than $300ns$, which means that 300 kbps data rate communication can be supported by CCT without significant ISI degradation [8]. Also, thanks to the RTTS statistical convergence property proved and experimentally demonstrated in [7] for a line-shaped network of clusters, RTTS variance doesn't increase as the hop number increases, so that CCT can be used for broadcasting in large multi-hop networks.

### 2.3.2 Semi-Cooperative Spectrum Fusion

In SCSF, multiple sensor nodes transmit their analog scalar data simultaneously to a receiver in response to a beacon [3]. The data is modulated by using frequency shift,

i.e., analog frequency modulation. Since the nodes transmit the modulated signal simultaneously, the spectrum of the super-imposed signal includes the information on the scalar values. So the receive node can estimate the average of the data by just taking the center of mass of the estimate of the received power spectral density. This scheme has the advantage that every node doesn't need to transmit exactly the same data, contrary to CCT. The different scalar information for each node can be transmitted through SCSF to the receiver, then the receiver can estimate the average of scalar values. Since the averaging is performed in the physical layer, it consumes a very small amount of energy and needs only a simple circuit.

# CHAPTER III

# COOPERATIVE ANALOG AND DIGITAL (CANDI) TIME SYNCHRONIZATION FOR LINE NETWORKS

In this chapter, we introduce the novel Cooperative Analog and Digital (CANDI) Time Synchronization Protocol. This protocol is the first TS method that combines analog and digital types of CT for range extension. First, we will describe the system model for the line network, then explain how the CANDI TS protocol performs.

## *3.1 System model*

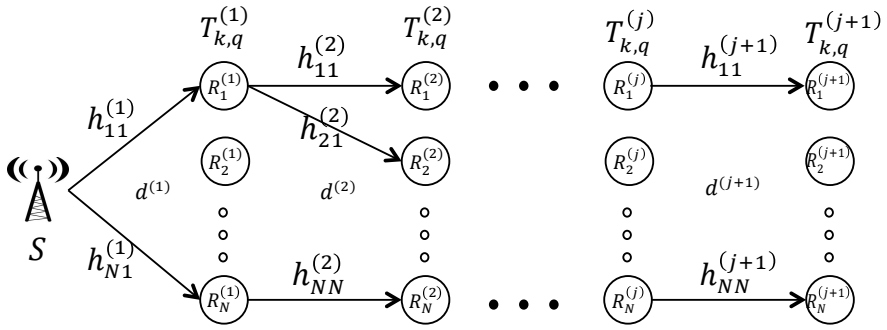



Figure 7: Multi-hop distributed concurrent cooperative transmission

In Figure 7, the line network topology for CT is given. $R_k^{(j)}$ denotes the $k^{th}$ node in the $j^{th}$ cluster, and all the nodes in the same cluster are co-located.[1] $R_k^{(j)}$ has the local clock $T_{k,q}^{(j)}$, where $q$ is the phase number, which will be described in Section 3.2. The purpose of TS is to make $T_{k,q}^{(j)}$ be the same time as the clock of the source node, so we define the error of local clock $T_{k,q}^{(j)}$ as $e_{k,q}^{(j)} = T_{k,q}^{(j)} - t$, where $t$ is the source node's clock. The $j^{th}$ and $j-1^{th}$ clusters are separated by the distance $d^{(j)}$ in meters

[1]Here, "co-located" means the nodes in a cluster are close enough that they all have the same pathloss to nodes in the next cluster, but they are separated sufficiently (e.g., by at least a quarter wavelength in a rich multipath environment) to have uncorrelated fading channels.

corresponding to the propagation time $\tau^{(j)} = d^{(j)}/(3 \times 10^8)$. We assume the distance $d^{(j)}$ is large enough that only adjacent clusters can receive the transmitted signals.

Let $h_{kr}^{(j)}$ be the complex Rayleigh flat faded channel gain between $R_r^{(j-1)}$ and $R_k^{(j)}$, such that $E\{|h_{kr}^{(j)}|^2\} = 1$ and $h_{kr}^{(j)}$ is constant for the duration of time-synchronization.

We assume that each node in a cluster transmits in a distinct diversity channel, and that the receivers are capable of diversity combining [8]. $SNR_1$ denotes the average received SNR when $R_k^{(1)}$ receives signals from the source node and $SNR_j$ denotes the average received SNR when the node $R_k^{(j)}$ for $j \geq 2$ receives signals from any one node in the previous cluster. We assume all radios have the same transmit power, antenna gains, and the receiver noise power. Therefore if $\alpha$ is the pathloss exponent, then $SNR_j = \left(\frac{d^{(1)}}{d^{(j)}}\right)^{\alpha} SNR_1$. To simplify our analysis, we assume that all intended receivers are able to decode digital packets without error; we use SNR to analyze timing error. We note that $SNR_j$ is much less than $SNR_1$, since CCT provides significant range extension [16].

## *3.2 Description of CANDI Time Synchronization Protocol for Line Networks*

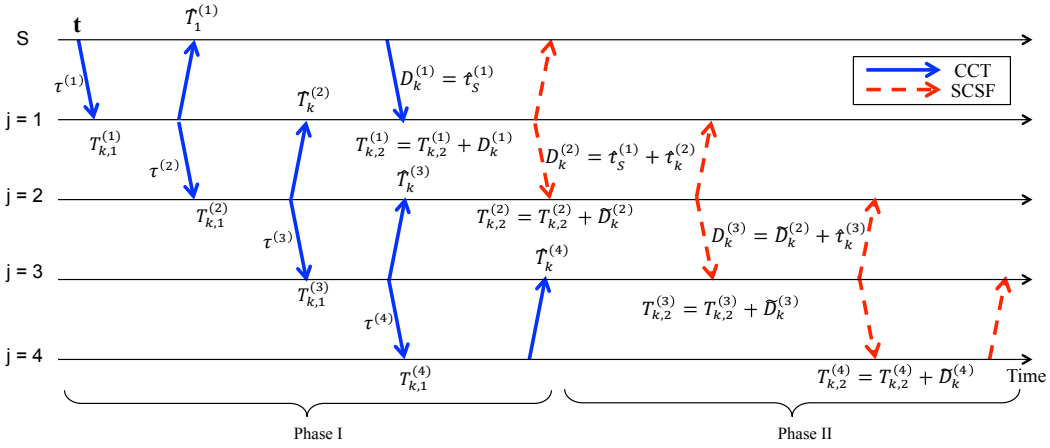



Figure 8: CANDI synchronization protocol

We begin this section with a high-level description of the CANDI protocol, then follow with detailed mathematical descriptions of each of Phase I and II, including

estimates and expressions for the time error distributions for each hop.

Figure 8 shows a timing diagram that we will use to describe the CANDI protocol. In that figure, the horizontal axis indicates time and the vertical axis indicates cluster number where $S$ means source. The solid arrows indicate a digital CCT and the dashed arrows indicate a SCSF transmission. The procedure starts with Phase I when the source broadcasts a TS packet, which includes the source clock time $t$ as digital data, to the nodes in the $1^{st}$ cluster. When the nodes $R_k^{(1)}$ for k=1,2, ...,N, receive the TS packet, they set their local clocks to the source time. They wait for the duration of $T_{proc}$ and broadcast the received TS packet simultaneously to the nodes in the next cluster by using CCT, where we can assume $T_{proc} = 0$, since $T_{proc}$ is deterministic. The nodes in the next cluster, $R_k^{(2)}$ for k=1,2, ...,N, receive the TS packet, set their local clocks and transmit the packet in the same way that the previous nodes $R_k^{(1)}$ do. In this way, the digital time stamp is propagated down the network. We note that Phase I is very similar to an opportunistic large array (OLA) broadcast [27]. The difference is that we constrain the cooperating nodes to be a co-located group, and we assume all nodes in the group decode without error. The timing errors will be similar to those measured in the "ping pong" experiment in [8].

The source node overhears the TS packet sent by CCT from $R_k^{(1)}$. By using the center of mass estimation of the SOP [8], where the best linear unbiased estimator (BLUE) is used, the source node can estimate the propagation time $\tau^{(1)}$ similarly to TPSN [11]. Then the source node transmits the estimated propagation time $D^{(1)} = \hat{\tau}^{(1)}$ digitally to the nodes in the $1^{st}$ cluster, which adjust their local clocks more precisely by adding this estimated propagation time to their local clocks.

Similarly, when the nodes in the $j^{th}$ cluster ($j \geq 2$) broadcast the TS packets in Phase I, every node in the $j-1^{th}$ cluster also can overhear the packets by the way of CCT, and estimate the propagation time $\tau^{(j)}$ as the source node does. The estimated propagation time at the $k^{th}$ node in the $j^{th}$ cluster is expressed $\hat{\tau}_k^{(j)}$. These $\hat{\tau}_k^{(j)}$ vary

from node to node within the $j^{th}$ cluster because of other non-deterministic time error sources, so that the nodes in the $j^{th}$ cluster cannot transmit $\hat{\tau}_k^{(j)}$ by CCT.

To transmit slightly different time information, we apply the Semi-Cooperative Spectrum Fusion (SCSF) [3] method for our analog stage. In SCSF, multiple sensor nodes transmit their scalar data simultaneously to a receiver in response to a beacon. The data is modulated by using frequency shift, i.e., analog linear frequency modulation [3]. Since the nodes transmit the modulated signal simultaneously, the spectrum of the super-imposed signal includes the information on the scalar values. So the receive node can estimate the average of the data by just taking the center of mass of the estimate of the received power spectral density [3]. We will apply SCSF to the stages in which a cluster of nodes needs to transmit an estimate of the time to the next cluster, however, each node in the cluster has a slightly different estimate because of estimation errors. SCSF eliminates the need for the cluster to reach consensus on the time, before it transmits.

This SCSF scheme is already studied in [3], where the author assumes that the fusion node is airborne, thus there exists Line of Sight (LOS) between the sensor nodes and the fusion node. However, since we assumed in Section 3.1 that the network structure is on the two-dimensional large network, we will apply SCSF in the absence of LOS, assuming Rayleigh fading. We have studied the performance of SCSF on the non-LOS channel, and found that it is effective at reducing estimation errors while simultaneously providing diversity gain.

So the nodes $R_k^{(j)}$ for k=1, 2, ..., N broadcast $D_k^{(j)} = \tilde{D}_k^{(j-1)} + \hat{\tau}_k^{(j)}$ by using SCSF to the nodes in the $j^{th}$ cluster, where $\tilde{D}_k^{(j-1)}$ is the SCSF estimated average of $D_n^{(j-1)}$ for n=1, 2, ..., N sent from $R_n^{(j-1)}$, that is obtained by

$$\tilde{D}_k^{(j-1)} = \frac{1}{h}\left(\frac{\sum_{n=1}^{N_{win}} v_n |\mathcal{F}_k(n)|^2}{\sum_{n=1}^{N_{win}} |\mathcal{F}_k(n)|^2} - f_c\right), \tag{1}$$

where $N_{win}$ is FFT size, $v_n$ is the $n^{th}$ DFT frequency, $\mathcal{F}_k(n)$ is $n^{th}$ DFT value of the received signal, $f_c$ is the carrier frequency and $h$ is the modulation index [3].

### 3.2.1 Phase I Detailed Description

The digital phase, or Phase I, of CANDI depends on both the Start of Packet (SOP) and End of Packet (EOP) estimates, which are computed by a node, when a packet is received. Two preambles, one at the beginning of the packet, and one at the end, are provided for these estimates, respectively. The SOP is used to provide a time reference for scheduling the time when the node will transmit the relayed packet to allow adequate time, $T_{proc}$, for packet processing.

#### *3.2.1.1 First Hop*

If we assume that the time-stamp is attached to TS packet immediately before the node transmits[2], the send time error at the source node can be assumed to be negligible. Thus the error between the local clock $T_{k,1}^{(1)}$ and real time $t$ can be simply expressed as

$$e_{k,1}^{(1)} = w_{k1}^{(1)} + \tau^{(1)}, \tag{2}$$

where $w_{k1}^{(1)}$ is the SOP estimate error at the $R_k^{(1)}$ node occurring while receiving the packet from the source node $S$, and the true propagation delay $\tau^{(1)}$ is strictly part of the error at this stage. The SOP estimate error can be expressed mathematically as in [12],

$$w_{k1}^{(j)} = \frac{x_{k1}^{(j)}}{|h_{k1}^{(j)}|}\sqrt{\frac{m}{SNR_1}}, \tag{3}$$

where $m$ is a constant that depends on the SOP time detection method, and $x_{k1}^{(j)}$ is a standard normal random variable. We can easily note that the variance of $w_{k1}^{(j)}$ is $m/SNR_{k1}^{(j)}$, where $SNR_{k1}^{(j)} = |h_{k1}^{(j)}|^2 SNR_1$.

[2]In [7], CCT is implemented in a Software Defined Radio (SDR) consisting of RF-daughterboard (RFX-2400), an Universal Software Radio Peripheral (USRP1) board, a personal computer (PC), and the GNU radio software. The time-stamp is attached to TS packet when the packet is passed from USRP1 board to RFX-2400.

#### *3.2.1.2 Second Hop*

Before the nodes in the $1^{st}$ cluster transmit the received TS packet, they wait for a period of deterministic time $T_{proc}$, to fire simultaneously. However, $T_{proc}$ for each node is not the same in reality, since all the nodes have slightly different clock rates. Thus the clock rates need to be compensated by using two preambles located prior and behind the time-stamp in the TS packet. However, those compensations are not perfect due to the SOP and EOP errors, therefore $T_{proc}$ of the node $R_k^{(1)}$ has the error $\xi_k^{(1)}$, which can be expressed as $\xi_k^{(1)} = Tproc \times f_s \times \frac{w_{k1}^{(1)} + \phi_{k1}^{(1)}}{B_s}$, where $\phi_{k1}^{(1)}$ is EOP error, $B_s$ is the number of samples per packet and $f_s$ is the source node's clock frequency. The detailed derivation is given in Appendix.

For the $2^{nd}$ hop, each $R_k^{(2)}$ for k=1, 2, ...,N receives the N copies of the same messages in different diversity channels, such that each signal is sent from $R_r^{(1)}$ node for r=1, 2, ...,N through channel $h_{kr}^{(2)}$. When each node combines these signals to achieve SNR advantage in CCT, the time errors are also combined. Before combining these time errors, we need to define the local error between the $R_r^{(1)}$ transmitter and the $R_k^{(2)}$ receiver, which can be expressed as

$$\hat{e}_{kr}^{(2)} = e_{r,1}^{(1)} + w_{kr}^{(2)} + \xi_r^{(1)} + \tau^{(2)}, \tag{4}$$

where $w_{kr}^{(2)}$ is the local SOP estimate error for the $kr^{th}$ link.

Then each receive node can apply BLUE to combine the local errors to minimize the combined error as in [8]. Thus, the combination of local clock errors of $T_{k,1}^{(2)}$ can be written as

$$\begin{aligned} e_{k,1}^{(2)} &= \sum_{r=1}^{N} \alpha_{kr}^{(2)} \hat{e}_{kr}^{(2)} \\ &= \sum_{r=1}^{N} \alpha_{kr}^{(2)} e_{r,1}^{(1)} + \sum_{r=1}^{N} \alpha_{kr}^{(2)} w_{kr}^{(2)} + \sum_{r=1}^{N} \alpha_{kr}^{(2)} \xi_r^{(1)} + \tau^{(2)}, \end{aligned} \tag{5}$$

where $\alpha_{kr}^{(2)} = \frac{|h_{kr}^{(2)}|^2}{\sum_{r=1}^{N} |h_{kr}^{(2)}|^2}$ are BLUE coefficients. We note that the BLUE estimate

is approximated automatically in [8] when the center-of-mass is found at the output of the noncoherent combiner.

#### *3.2.1.3 Other Hops*

When the node $R_k^{(j)}$ for $j \geq 3$ receives the TS packet, it also sets the received time as its local time and broadcasts the packet simultaneously. The local error between the $r^{th}$ transmitter in the $j-1$ cluster and the $k^{th}$ receiver node in $j$ cluster can be expressed as

$$\hat{e}_{kr}^{(j)} = e_{r,1}^{(j-1)} + w_{kr}^{(j)} + \xi_r^{(j-1)} + \tau^{(j)}. \tag{6}$$

Note that $\xi_r^{(j-1)}$ occurs at the node $R_r^{(j-1)}$ while waiting for $T_{proc}$ that is compensated by using the combined estimate of SOP and EOP, thus it can be expressed as $\xi_r^{(j-1)} = T_{proc} \times f_s \times \dfrac{\tilde{w}_r^{(j-1)} + \tilde{\phi}_r^{(j-1)}}{B_s}$, where $\tilde{w}_r^{(j-1)} = \sum_{p=1}^{N} \alpha_{rp}^{(j-1)} w_{rp}^{(j-1)}$ and $\tilde{\phi}_r^{(j-1)} = \sum_{p=1}^{N} \alpha_{rp}^{(j-1)} \phi_{rp}^{(j-1)}$, respectively.

Therefore, the combination of local clock errors of $T_{k,1}^{(j)}$ can be expressed as

$$\begin{aligned} e_{k,1}^{(j)} &= \sum_{r=1}^{N} \alpha_{kr}^{(j)} \hat{e}_{kr}^{(j)} \\ &= \sum_{r=1}^{N} \alpha_{kr}^{(j)} e_{r,1}^{(j-1)} + \sum_{r=1}^{N} \alpha_{kr}^{(j)} w_{kr}^{(j)} + \sum_{r=1}^{N} \alpha_{kr}^{(j)} \xi_r^{(j-1)} + \tau^{(j)}, \end{aligned} \tag{7}$$

similarly as the $2^{nd}$ hop case.

### 3.2.2 Phase II Detailed Description

When the source node overhears the packet sent from the $1^{st}$ cluster, this can be seen as similar to the pair-wise synchronization as in TPSN [11]. Thus the source node can have the estimated propagation time, $\hat{\tau}_1^{(1)} = \dfrac{\hat{T}_1^{(1)} - t}{2}$, where $\hat{T}_1^{(1)}$ is the real time when the source node overhears. More specifically, $\hat{\tau}_1^{(1)} = \dfrac{\sum_{r=1}^{N} \alpha_{1r}^{(0)'} \hat{e}_{1r,1}^{(0)'}}{2}$ can be expressed as

$$\hat{\tau}_1^{(1)} = \left( \sum_{r=1}^{N} \alpha_{1r}^{(0)'} w_{r1}^{(1)} + \sum_{r=1}^{N} \alpha_{1r}^{(0)'} w_{1r}^{(0)'} + \sum_{r=1}^{N} \alpha_{kr}^{(0)'} \xi_r^{(1)} \right) /2 + \tau^{(1)}, \tag{8}$$

where $\hat{e}_{1r}^{(0)'} = e_{r,1}^{(1)} + w_{1r}^{(0)'} + \xi_r^{(1)} + \tau^{(1)}$ is the reverse direction local error along the path between $R_r^{(1)}$ and the source node. We have $\alpha_{1r}^{(0)'} = \frac{|h_{1r}^{(0)'}|^2}{\sum_{r=1}^{N} |h_{1r}^{(0)'}|^2}$ as the reverse direction BLUE coefficients, and $w_{1r}^{(0)'}$ is the reverse direction local SOP estimate error from the node $R_r^{(1)}$. We can note that $\hat{\tau}_1^{(1)} \sim \mathcal{N}(\tau^{(1)}, \sigma^2_{\hat{\tau}_1^{(1)}})$, since $w_{r1}^{(1)}$, $w_{1r}^{(0)'}$, and $\xi_r^{(1)}$ are all normally distributed with zero mean and the linear combination of normal variables is also a normal random variable.

Next, the source node transmits $D_k^{(1)} = \hat{\tau}_1^{(1)}$ to the nodes in the $1^{st}$ cluster, and the $R_k^{(1)}$ modify their times, i.e. from $T_{k,1}^{(1)}$ to $T_{k,2}^{(1)} = T_{k,1}^{(1)} + \hat{\tau}_S^{(1)}$. Thus the local clock $T_{k,2}^{(1)}$ at Phase 2, i.e., $e_{k,2}^{(1)} = e_{k,1}^{(1)} - \hat{\tau}_1^{(1)}$, has the TS error,

$$e_{k,2}^{(1)} = w_{k1}^{(1)} - \left( \sum_{r=1}^{N} \alpha_{1r}^{(0)'} w_{1r}^{(1)} + \sum_{r=1}^{N} \alpha_{1r}^{(0)'} w_{1r}^{(0)'} + \sum_{r=1}^{N} \alpha_{kr}^{(0)'} \xi_r^{(1)} \right) /2, \tag{9}$$

where the propagation time $\tau^{(1)}$, which is the deterministic unknown error, has been eliminated.

Also, because the nodes in the $1^{st}$ cluster overhear the TS packet sent from the nodes in the $2^{nd}$ cluster as the source node does, the node $R_k^{(1)}$ is also able to calculate the estimated propagation time of $2^{nd}$ hop by $\hat{\tau}_k^{(2)} = \frac{\hat{T}_k^{(2)} - T_{k,1}^{(1)}}{2}$, where $\hat{T}_k^{(2)}$ is the estimated time when the $R_k^{(1)}$ overhears the combined packet. Specifically, $\hat{\tau}_k^{(2)}$ can be expressed by

$$\begin{aligned} \hat{\tau}_k^{(2)} &= \left( \sum_{r=1}^{N} \alpha_{kr}^{(1)'} \hat{e}_{kr}^{(2)'} - e_{k,1}^{(1)} \right) /2 \\ &= \left( \sum_{r=1}^{N} \alpha_{kr}^{(1)'} (e_{r,1}^{(2)} + w_{kr}^{(1)'} + \xi_r^{(2)} + \tau^{(2)}) - e_{k,1}^{(1)} \right) /2. \end{aligned} \tag{10}$$

Similarly, we can note that $\hat{\tau}_k^{(2)}$ is also normally distributed, i.e., $\hat{\tau}_k^{(2)} \sim \mathcal{N}(\tau^{(2)}, \sigma^2_{\hat{\tau}^{(2)}})$, since $w_{kr}^{(j)}$, $w_{kr}^{(j)'}$ and $\xi_k^{(j)}$ for $j \geq 1$, $1 \leq (k, r) \leq N$ are normally distributed with zero

mean and

$$\begin{aligned} E[\hat{\tau}_k^{(2)}] &= E\left[\left(\sum_{r=1}^{N}\alpha_{kr}^{(1)'}(e_{r,1}^{(2)}+w_{kr}^{(1)'}+\xi_r^{(2)}+\tau^{(2)})-e_{k,1}^{(1)}\right)/2\right] \\ &= \left(E\left[\sum_{r=1}^{N}\alpha_{kr}^{(1)'}e_{r,1}^{(2)}\right]+\tau^{(2)}-E\left[e_{k,1}^{(1)}\right]\right)/2 \\ &= \left(\sum_{k=1}^{2}\tau^{(k)}+\tau^{(2)}-\tau^{(1)}\right)/2=\tau^{(2)}. \end{aligned} \tag{11}$$

Then, $R_k^{(1)}$ transmits $D_k^{(2)}=\hat{\tau}_S^{(1)}+\hat{\tau}_k^{(2)}$ to $R_k^{(2)}$ using SCSF, since $D_k^{(2)}$ for k=1, 2, ..., N are slightly different due to that $\hat{\tau}_k^{(2)}$ is different for each k. As shown above, we note that $D_k^{(2)}\sim\mathcal{N}(\tau^{(1)}+\tau^{(2)},\sigma^2_{D_k^{(2)}})$. Thus the information on $D_k^{(2)}$ for k=1, 2, ..., N are all super-imposed on the transmitted signal frequency at the physical layer and the $R_k^{(2)}$ can estimate the average propagation time $\tilde{D}_k^{(2)}$ by using Equation (1). Then the node $R_k^{(2)}$ adjust its local clock as $T_{k,2}^{(2)}=T_{k,1}^{(2)}+\tilde{D}_k^{(2)}$.

These processes are carried out throughout the whole network, eventually then every node has the local clock synchronized to the source clock. This CANDI scheme requires only 2 waves of broadcasting the time information, the time required to synchronize the whole network is in proportion to $2\times H$ where H is the total hop number, which is shorter than the non-CT hop number.

### *3.3 Simulation Result*

In this section, MATLAB simulations are given to show the performance of the CANDI TS protocol. We use the constant $m=6.039\times10^{-9}$ as in [12] and set $SNR_1=25dB$, $SNR_j=10dB$, and $\alpha=3$. Let the propagation time $\tau^{(j)}$ for the distance of $d_1$ and $d_2$ be $0.1\mu sec$ and $0.33\mu sec$, respectively. The data rate is 64kHz and the sample rate is 1MHz. We use $T_{proc}=20ms$, which is sufficient for a block buffer of 4kB [8]. In SCSF, we use the carrier frequency[3] $f_c=512kHz$, the sampling frequency $f_s=128kHz$, and FFT size $N_{win}=128$. To evaluate the performance of

[3]We note that theoretically the value of $f_c$ doesn't matter.

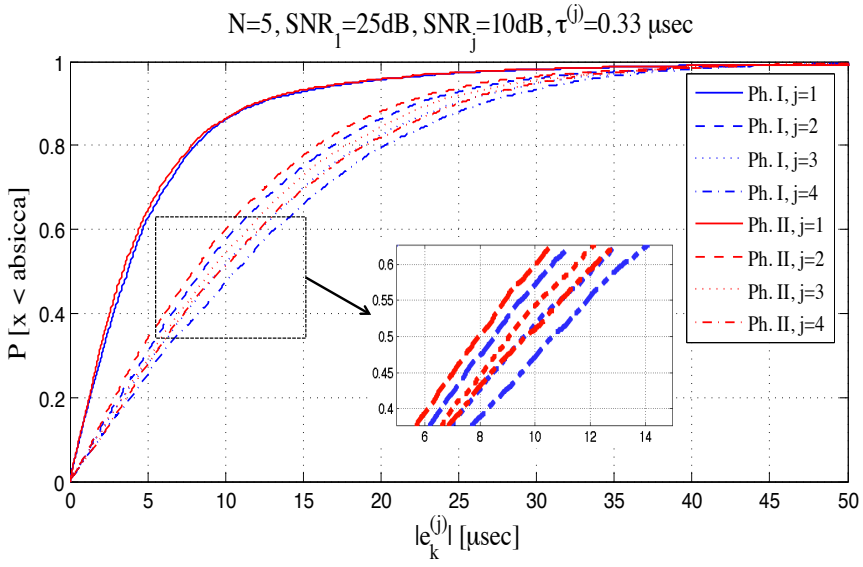



Figure 9: CDF of absolute TS error, $|e_k^{(j)}|$, for CANDI

TS protocol, we use $RMSE(j) = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}|e_k^{(j)}|^2}$.

### 3.3.1 Performance of CANDI

Figure 9 provides the Cumulative Density Function (CDF) of the clock error $|e_k^{(j)}|$, for $1 \leq j \leq 4$ and $N = 5$, of CANDI. We observe that the median error ranges from $8\mu s$ to $11\mu s$, and increases slightly with hop index. Note that Phase II reduces the TS error as the amount of the propagation time approximately.

Figure 10 shows the RMSE performance with various numbers of nodes per cluster. We can note that the TS error decreases as the number of nodes in the cluster increases, because the quality of the average improves as the number of waveforms being averaged increases, as in a sample mean.

### 3.3.2 Comparison between CANDI and TPSN

In the CANDI simulation, we note the distance between the source node and the $10^{th}$ cluster is $d_1 + 9 \times d_2 = 29.44 \times d_1$. The number of nodes for each cluster is $N = 4$, thus the total number of nodes is 40 for 10 clusters. For TPSN, which is the popular TS protocol, we use the same total number of nodes, same network length, and the

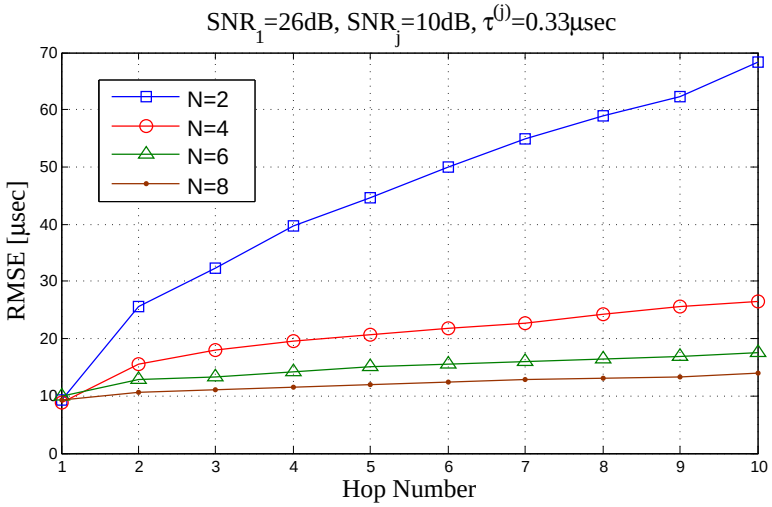



Figure 10: CANDI RMSE by cluster size

same transmit power per nodes as CANDI. However, the nodes are equally spaced for TPSN, with separation $\frac{29.44d_1}{40} = 0.736d_1$, such that the SNR received from an adjacent node is $SNR_{TPSN} = SNR_1(d_1/d_{TPSN})^3 = 28.99dB$.

In TPSN, each node sends REQ TS packet to neighboring nodes, then the randomly chosen node replies with the ACK packet if it receives the REQ packet correctly (we assume the SNR threshold is 10dB as in [16]) and has the time information, then the two nodes perform the TS process.

Firgure 11 shows the comparison between CANDI and TPSN. From this result, we can observe that the TS error of CANDI increases with a much smaller increment as the distance increases compared with TPSN. At the point of $30 \times d_0$, the RMSE of TPSN is about 1.5 times that of CANDI.

## *3.4 Conclusion*

In this chapter, we describe the Cooperative Analog and Digital (CANDI) TS protocol for large multi-hop line networks. The simulation results show that the averaging and range extension benefits of cooperative transmission (CT) lead to a large improvement over the popular non-CT-based TPSN.

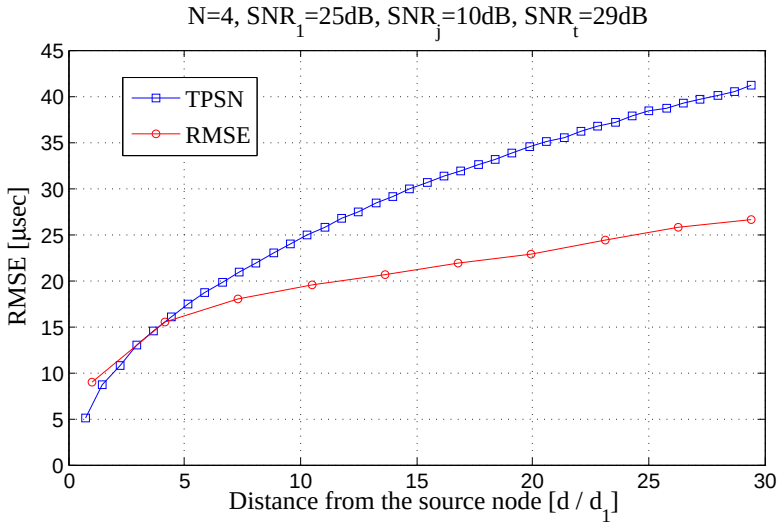



Figure 11: Comparison of CANDI and TPSN (Both networks have the same energy consumption)

# CHAPTER IV

# COOPERATIVE ANALOG AND DIGITAL (CANDI) TIME SYNCHRONIZATION FOR TWO-DIMENSIONAL NETWORKS

In this chapter, we evaluate the performance of CANDI TS in two-dimentional (2-D) networks. First we describe the system model and assumptions for these networks, then explain the high-level description of CANDI for 2-D networks. Then, the simulation results are given. The mathematical description of CANDI is not explained in this chapter, since the basic mechanism of CANDI TS protocol in 2-D networks is almost the same as the case of the line network.

## *4.1 System Model*

Let consider the network with the number of nodes N, and they are uniformly distributed in a square area $A$. Let $R_k$ denote the $k^{th}$ node, and we assume $R_k$ has the local clock $T_{k,p}$, where $p$ is the phase number. Since the purpose of TS is to make the local clock same to the time of the source node, the time error of $R_k$ can be defined $e_{k,p} = T_{k,p} - t$, where $t$ is the reference time, i.e., the clock of the source node.

We assume the source node is located at the origin, and all the nodes are assumed to have the same transmit power. Let $h_{kl}$ denote the channel from $R_l$ to $R_k$. All the channels are assumed to be Rayleigh flat faded, such that $E\{|h_{kl}|^2\} = 1$, uncorrelated and static for the duration of TS. By assuming the average received SNR depends only on the distance between the nodes with the pathloss exponent $\alpha$, we can have the SNR from the node $R_l$ to $R_k$ can be expressed by $SNR_{kl} = |h_{kl}|^2(\frac{d_{(k,l)}}{d_0})^{\alpha} SNR_0$, where $d_{(k,l)}$ is the distance between those two nodes and $d_0$ is the reference distance

having the $SNR_0$. Here, $d_{(k,l)}$ corresponds to the propagation time $\tau_{(k,l)} = \frac{d_{(k,l)}}{3\times10^8}$, taken to transmit a signal from $R_k$ to $R_l$.

In contrast to the assumption of the last chapter, where the same number of nodes decode in every cluster, in this chapter, the number of nodes that decode in each cluster is not pre-determined as in OLAs broadcasting [18] discussed in Section 2.3.1. When the nodes receive and decode the TS packet, they refer to the cluster level contained in the packet and designate themselves to the specified level. This procedure will be described in detail in the following section.

To achieve the maximum micro-diversity gain in a virtual array, the channels between relays and a receiver should share the same pathloss and shadowing, but each should have uncorrelated multipath fading. Note that this condition is satisfied in the line networks of the previous chapter. However in the 2-D networks of this chapter, since the pathloss of each link is different due to the different distance, some nodes' contributions to the diversity gain are very small, which would lead to an effective diversity order less than the number of relays participating in CT [16].

## *4.2 Description of CANDI Time Synchronization Protocol for Two-dimensional Networks*

In this section, we describe how the CANDI TS protocol performs in 2-D networks. CANDI consists of two phases. In Phase I, the TS packet made at the source node is distributed though the whole network using CCT, and in Phase II, the estimated propagation delay is transmitted by using SCSF.

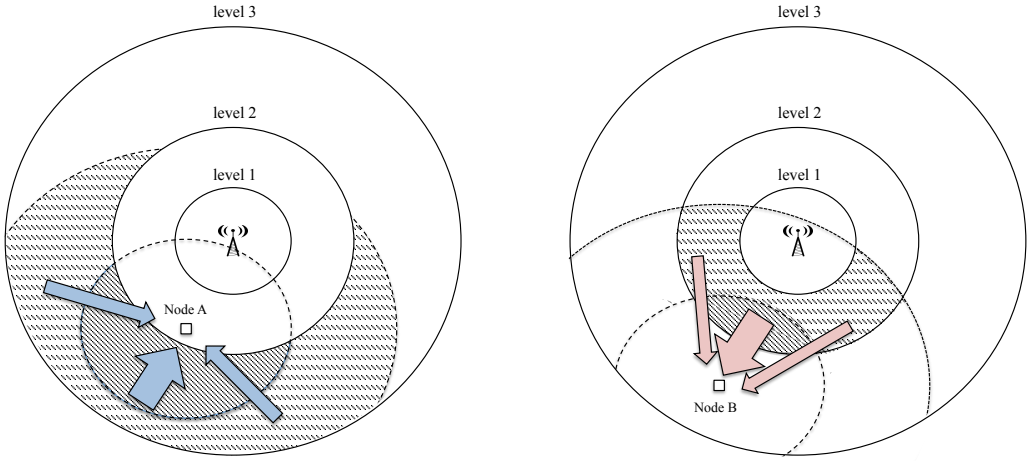
Once the Phase I of CANDI begins, the source node broadcasts the TS packet including the time information and the level of cluster, i.e. $level = 1$. If neighboring nodes can receive the packet without error (we assume the SNR threshold to decode the packet without error is 10dB as in [16]), they designate themselves as the $1^{st}$ cluster and set their local clocks by using the received time information contained in the TS packet. Then, they increase the cluster number of the TS packet, i.e.,

$level = 2$, and broadcast the TS packet using CCT. The other nodes farther away than $1^{st}$ cluster can receive the TS packet if they have the aggregate SNR higher than 10dB, i.e., $\sum_{i \in \mathbf{C}^{(1)}} SNR_{i\ (dB)} \geq 10dB$, where $\mathbf{C}^{(1)}$ is the node numbers of $1^{st}$ cluster [16]. As in the $1^{st}$ cluster, the nodes designate to the $2^{st}$ cluster by themselves, and set their local clocks by using the relayed TS packet and increase the cluster number of TS packet to $level = 3$. Here, the nodes that already belong to $1^{st}$ cluster don't change their local clocks with the TS packet. Successively, the nodes relay the TS packet using CCT as the $1^{st}$ cluster's nodes did. This process is carried out through the whole network. Note that Phase I is very similar to OLA broadcasting [18], where the clusters are not pre-determined.

Phase II is composed of two parts. The first part is the overhearing of the digital transmissions of Phase I. The second part is the analog transmission and reception using SCSF. In both parts, the estimates of propagation time that are done using the received signals are influenced more by the nearest nodes, because of the distributed nature of the clusters. We now explain this local influence for each part.

Regarding the first part of Phase II, we recall for the line network of the last chapter that the source node can overhear the TS packet sent from the $1^{st}$ cluster, and then estimate the propagation time by combining the SOP times of multiple packets. However, in the case of the 2-D network, the nodes in each cluster are not co-located; therefore the true propagation delays between the source node and the nodes in the $1^{st}$ cluster are all different. Therefore, the source's estimate of the round-trip propagation time is influenced most by the nearest nodes.

This feature of nearest node influence also impacts estimates in later hops. In 2-D OLA broadcasts [30], each cluster has the shape of a ring around the source node and the thicknesses of the rings increase with hop index if the conditions for successful OLA broadcasts are met: $SNR_{th} \leq \pi(ln2)\rho P_r$, where $\rho$ is the density of




(a) The relative amount of effect on the estimating the propagation delay

(b) The estimated propagation delay transmitted using SCSF

Figure 12: Local clusterization using the property of combining of CCT and SCSF

nodes and $P_r$ is the relay transmit power.[1] Therefore, when the nodes in the present cluster overhear the TS packet sent from the next cluster as shown in Figure 12a, strictly, the TS packets that are fired from the nodes located at the opposite side of the ring also arrive. However, when the node estimates the propagation time, the local SOP times for each channel are combined using BLUE coefficients [8] as described Section 3.2.1.2. Note that BLUE coefficients depend entirely on the channel gains, which are mainly depending on the distance between the nodes. Therefore, the neighboring nodes dominate the estimation of the propagation delay, since they have usually the highest SNRs. For example, Figure 12a shows that Node A in the $2^{nd}$ cluster receives multiple TS packets from the $3^{rd}$ cluster. The width of the arrow means the amount of influence on the estimation of Node A. We can note that the nodes in $3^{rd}$ cluster near from Node A can have a large influence on Node A, so that Node A's estimated propagation time can be almost the estimation of the propagation time between Node A and the 'local' cluster of nearest nodes.

[1] In [30], they assume a continuum of nodes in the network, i.e., $\rho \to \infty$.

In the second part of Phase II, when the nodes transmit the estimated propagation time to the next cluster using SCSF, the receive node estimates the average propagation time just by taking the center of mass of the estimate of the received power spectrum. Thus, the estimated average by SCSF is also affected by the SNRs between the transmit nodes and the receive node. For example, in Figure 12b, Node B receives the super positioned spectrum from the $2^{nd}$ cluster. Note that the near nodes have a large amount of influence on the Node B's estimation as the width of arrows mean.

As we will see next in the simulation results, CANDI gives good performance in terms of mean squared timing error, in spite of the "local" effects discussed above.

## *4.3 Simulation Result*

In this section, MATLAB simulations are given to show the performance of CANDI TS protocol for disk networks. We use the constant $m = 6.039 \times 10^{-9}$ and set $SNR_0 = 10dB$ for the distance $d_0 = 40m$, and $\alpha = 3$. The propagation time between the node $R_i$ and $R_j$ is $\tau_{(i,j)} = \frac{d_{(i,j)}}{3\times10^8}$, where $d_{(i,j)}$ is the distance between those two nodes. The data rate and the sample rate is is 64kHz, 1MHz, respectively. We use $T_{proc} = 20ms$ for CCT, and the carrier frequency $f_c = 512kHz$, the sampling frequency $f_s = 128kHz$, and FFT size $N_{win} = 128$ for SCSF.

### 4.3.1 Comparison between CANDI and TPSN

Table 1: The number of nodes per hop (N=300, A = $300 \times 300m^2$)

| Scheme | Hop Number | | | | | | | | | Synchronized Node (%) | Avg. Max Hop |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $1^{st}$ | $2^{nd}$ | $3^{rd}$ | $4^{th}$ | $5^{th}$ | $6^{th}$ | $7^{th}$ | $8^{th}$ | $9^{th}$ & More | | |
| TPSN | 5% | 8% | 11% | 15% | 17% | 15% | 12% | 8% | 9% | 100% | 11.01 |
| CANDI | 5% | 33% | 61% | 1% | | | | | | 100% | 3.49 |

The Figure 13 shows the performance comparison between CANDI and TPSN in 2-D uniformly distributed networks. In this simulation, we assume that 300 nodes are distributed in various sizes of network. To evaluate the accuracy for each protocol,

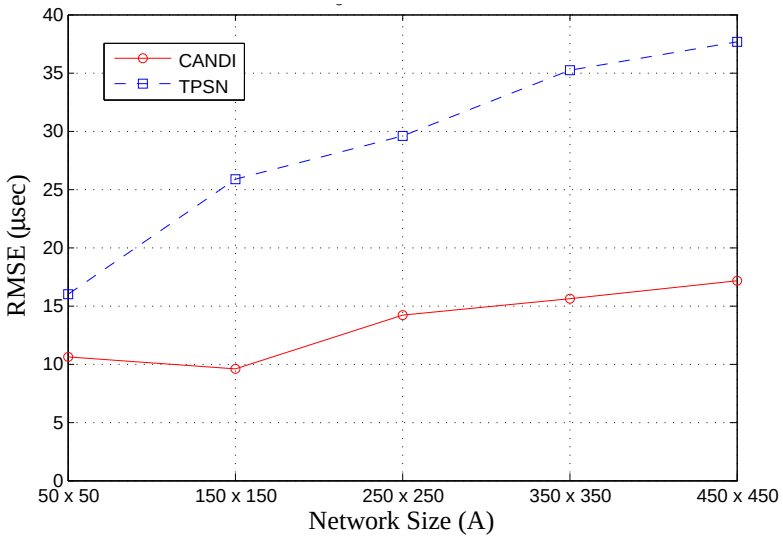



Figure 13: Comparison of CANDI and TPSN in 2-D networks

we use $RMSE = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}|e_k|^2}$.

As shown in Figure 13, we can note that the RMSE of TPSN is larger than CANDI for all the network sizes. There are two reasons why CANDI has better performance than TPSN. First, the numbers of nodes per each hop of TPSN and CANDI are given in Table 1 for the case of $N = 300$ and $A = 300 \times 300m^2$. We can note that the average maximum hop to cover the network when using CANDI is much less than the case of TPSN, which means that the advantage of range extension of CT occurs in CANDI. By reducing the total hop number to cover the networks, CANDI can reduce the TS error over the whole network. Second, in CANDI, since the number of nodes per cluster increases as the hop level increases as OLA broadcasts [18], the quality of average also improves as the cluster level increases.

### 4.3.2 The Ratio of Synchronized Nodes

When small numbers of nodes are distributed over areas that are large compared to their transmission range (i.e., when the network is sparse), the TS packet often fails to be relayed through the whole network, even though we use OLAs. The Figure 14 shows the percentage of synchronized nodes with the various number of node and

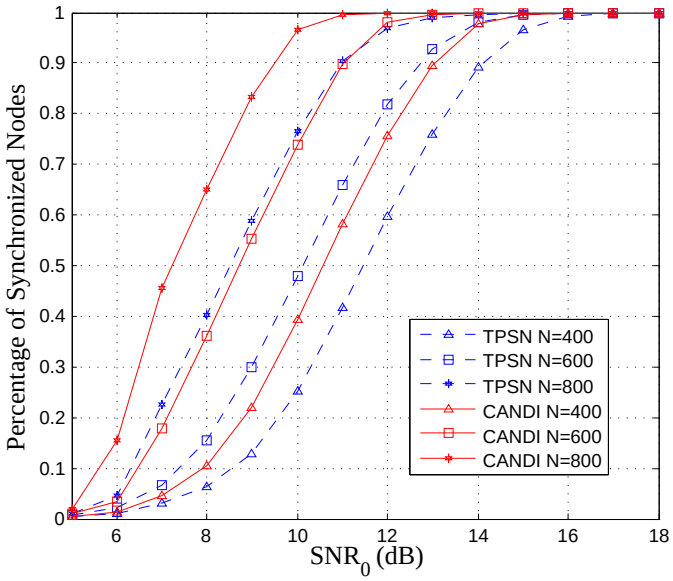



Figure 14: The rate of synchronized node (%)

the transmission power for the same network size $A = 900 \times 900m^2$. To change the transmit power of nodes, we change the reference received power $SNR_0$, which is the average received power at the reference distance $d_0 = 40m$. From the result, we can observe that the ratio of synchronized nodes in CANDI is higher than TPSN for all the cases. This result proves that, even when the CANDI doesn't satisfy the broadcast condition suggested in [30] (i.e., the percentage of synchronized nodes is not 100%), CANDI still has the advantage of range extension, compared to the non-CT method. This is another advantage of CANDI that the nodes can receive the TS packet starting from the source node, while they cannot receive the TS packet with the non-CT method.

## *4.4 Conclusion*

In this chapter, we show that CANDI also has the better performance than TPSN in the 2-D networks. Although the nodes in each cluster are not co-located, the properties of combining method of CCT and SCSF, which depend on the distance, enable the node to estimate the average propagation time between local clusters. We

can also observe that the ratio of synchronization nodes in CANDI is higher than TPSN in any networks environments.

# CHAPTER V

# CONCLUSION AND FUTURE WORK

## *5.1 Conclusion*

In this thesis, we have proposed the new TS method named the cooperative analog and digital time synchronization (CANDI), which exploits the two types of CT, CCT and SCSF. By using the advantage of the range extension of CT, CANDI is able to reduce the hop number needed to cover the network, so that the TS error can be also reduced. Also, the inherent property of averaging the timing error of multiple TS packets sent from multiple transmitters decreases the TS error increment between the clusters. By using MATLAB simulations, we prove that CANDI has the better performance than TPSN for the both of line and 2-D networks.

## *5.2 Future Work*

Below we list possible research topics related to the content of this thesis.

1. Though simulation results can be used to verify CANDI scheme is scalable and feasible, mathematical analysis is still needed to prove how CCT and SCSF reduce the TS error accumulation compared to typical non-CT TS methods. This analysis will be crucial part of finding optimal solution for obtaining the minimum TS error.

2. Due to the other non-deterministic TS errors, which cannot be modeled for simulations, the implementation of TS protocol is essential part to conclude developing novel TS scheme. CCT and SCSF can be implemented on a Universal Software Radio Peripheral (USRP) and GNU radio, thus the experiment of CANDI is feasible.

# APPENDIX A

## DERIVATIVE OF SEND TIME ERROR

Let's assume following variables,

$f_s$ : the clock rate of the source

$f_r$ : the clock rate of the receiver

$B_s$ : the number of samples per packet at the source

$B_r$ : the number of samples per packet at the receiver

$T_p$ : the packet duration

$C$ : the clock number of $T_{proc}$ at the soucre ($C = T_{proc} \times f_s$)

By using two preambles locating at the first and last of the packet sent from source, the receiver can calculate the ratio of clock rates between the source and itself,

$$k = \frac{f_s}{f_r} = \frac{B_s}{B_r} = \frac{B_s}{(T_p + w + \phi) \times f_r}$$

where $w$ and $\phi$ is the SOP and EOP estimation error, respectively. The receiver needs to wait for $T_{proc}$, it can calculate the estimation of $T_{proc}$ as

$$\begin{aligned} \hat{T}_{proc} &= C \times \frac{1}{k f_r} \\ &= C \times \frac{T_p + w + \phi}{B_s} \\ &= T_{proc} + C \times \frac{w + \phi}{B_s} \end{aligned}$$

where $C$ is known to the receiver. Note that the second term is the error of $T_{proc}$, which is the send time error,

$$\xi = T_{proc} \times f_s \times \frac{w + \phi}{B_s}.$$

# REFERENCES


[1] Available at http://bbn.com/boomerang.

[2] Available at http://www.darpa.mil/ato/programs/SHM/.

[3] Akanser, A. and Ingram, M., "Semi-cooperative spectrum fusion (SCSF) for aerial reading of a correlated sensor field," *1st Intl. Conf. on Wireless VITAE*, pp. 732 –736, May 2009.

[4] Akyildiz, I. F. and Vuran, M. C., "Wireless sensor networks. chapter in WSN applications," *John Wiley and Sons, Inc.*

[5] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, pp. 102 – 114, Aug. 2002.

[6] C. body, H. Mertodimedjo, G. H. H. U. and Mieghem, P., "Analysis of end-to-end delay measurements in internet," *Passive and Active Measurements Workshop, Fort Collins, Co.*

[7] Chang, Y. J. and Ingram, M., "Convergence property of transmit time pre-synchronization for concurrent cooperative communication," *IEEE GLOBECOM*, pp. 1 –5, Dec. 2010.

[8] Chang, Y. J., Ingram, M., and Frazier, R., "Cluster transmission time synchronization for cooperative transmission using software-defined radio," *IEEE ICC*, pp. 1 –5, May 2010.

[9] Cook, D. and Das, S., "Smart environments: Technologicy, protocols and applications," *Wiley-Interscience*, 2005.

[10] Elson, J., Girod, L., and Estrin, D., "Fine-grained network time synchronization using reference broadcasts," *5th Symp. OSDI*, pp. 147–163, 2002.

[11] Ganeriwal, S., Kumar, R., and Srivastava, M. B., "Timing-sync protocol for sensor networks," *1st Intl. Conf. on Embedded Networked Sensor Systems*, pp. 138–149, 2003.

[12] Gao, Z., Chang, Y. J., and Ingram, M., "Synchronization for cascaded distributed mimo communications," *MILCOM*, pp. 387 –392, 3 2010.

[13] Hong, Y.-W. and Scaglione, A., "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 1085 – 1099, May 2005.

[14] Hu, A.-S. and Servetto, S. D., "A scalable protocol for cooperative time synchronization using spatial averaging," 2006, [Online] Available: http://arxiv.org/PScache/cs/pdf/0611/0611003v1.pdf.

[15] Huang, P.-H., Desai, M., Qiu, X., and Krishnamachari, B., "On the multihop performance of synchronization mechanisms in high propagation delay networks," *IEEE Trans. on Computers*, vol. 58, pp. 577 –590, May 2009.

[16] Jung, H., Chang, Y. J., and Ingram, M., "Experimental range extension of concurrent cooperative transmission in indoor environments at 2.4GHz," *IEEE MILCOM*, pp. 148 –153, Nov 2010.

[17] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and Turon, M., "Health monitoring of civil infrastructures using wireless sensor networks," *6th International Symposium on IPSN*, pp. 254 –263, April 2007.

[18] L. Thanayankizil, A. K. and Ingram, M., "Routing for wireless sensor networks with an opportunistic large array (ola) physical layer," *Ad Hoc and Sensor Wireless Networks, Special Issue on Sensor Technologies and Applications*, vol. 8, pp. 79 –117, Jan. 2009.

[19] Laneman, J. and Wornell, G., "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Information Theory*, vol. 49, pp. 2415 – 2425, Oct 2003.

[20] Leon-Garcia, A., "Probability and random processes for electrical engineering, 2nd ed.," *MA: Addison-Wesley*, 1993.

[21] Lin, C.-Y., Peng, W.-C., and Tseng, Y.-C., "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 5, pp. 1044 –1056, Aug. 2006.

[22] Maróti, M., Kusy, B., Simon, G., and Lédeczi, A., "The flooding time synchronization protocol," *2nd Intl. Conf. on Embedded networked sensor systems*, pp. 39–49, 2004.

[23] Mills, D., "Internet time synchronization: the network time protocol," *IEEE Trans. on Communications*, vol. 39, pp. 1482 –1493, Oct. 1991.

[24] Pagliari, R. and Scaglione, A., "Scalable network synchronization with pulse-coupled oscillators," *IEEE Trans. on Mobile Computing*, vol. 10, pp. 392 –405, march 2011.

[25] Papoulis, A., "Probability, random variables and stochastic processes, 3rd ed.," *New York: McGraw-Hill*, 1991.

[26] Puccinelli, D. and Haenggi, M., "Wireless sensor networks: applications and challenges of ubiquitous sensing," *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19 – 31, 2005.

[27] Scaglione, A. and Hong, Y.-W., "Opportunistic large arrays: cooperative transmission in wireless multihop ad hoc networks to reach far distances," *IEEE Trans. Signal Processing*, vol. 51, pp. 2082 – 2092, Aug 2003.

[28] Schor, L., Sommer, P., and Wattenhofer, R., "Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings," *1st ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys), Berkeley, California, USA*, Nov 2009.

[29] Schwiebert, L., G. S. and Weinmann, J., "Research challenges in wireless networks of biomedical sensors," *SIGMOBILE*, p. 151165, 2001.

[30] Sirkeci-Mergen, B., Scaglione, A., and Mergen, G., "Asymptotic analysis of multistage cooperative broadcast in wireless networks," *IEEE Trans. on Information Theory*, vol. 52, pp. 2531 – 2550, June 2006.

[31] Song, P., Li, K., Shan, X., and Qi, G., "Multi-hop based highly precise time synchronization protocol for ZigBee networks," *3rd IEEE Intl. Symp. on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications,*, pp. 1197 –1201, Oct 2009.

[32] Tyrrell, A., Auer, G., and Bettstetter, C., "On the accuracy of firefly synchronization with delays," *1st Intl. Symposium on ISABEL*, pp. 1 –5, Oct 2008.

[33] Wu, Y.-C., Chaudhari, Q., and Serpedin, E., "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, pp. 124 –138, Jan 2011.

[34] Xu, N., "A survey of sensor network applications," *Survey Paper for CS694a, Computer Science Department, University of Southern California.*